



FIB

Facultat d'Informàtica
de Barcelona

UNIVERSITAT POLITÈCNICA DE CATALUNYA

CONCEPTES AVANÇATS DE SISTEMES OPERATIUS

Departament d'Arquitectura de Computadors

Java 2 Micro Edition

Introducción a java para dispositivos móviles

(Seminaris de CASO)

Autors

David Chiner Benjuya

Antonio Joaquín Duro Joyanes

Ramón Peralta Sáez

J2ME

- Versión de la plataforma Java orientada a dispositivos de recursos limitados.



Configuraciones/Perfiles

- J2ME se basa en los conceptos de configuración y perfil.
- Objetivo \Rightarrow optimizar librerías y MV para cada grupo de dispositivos.
- Configuración:
 - Conjunto formado por la máquina virtual Java y una selección de paquetes de librerías.
 - Mínimo común denominador de características y clases para un grupo de dispositivos.

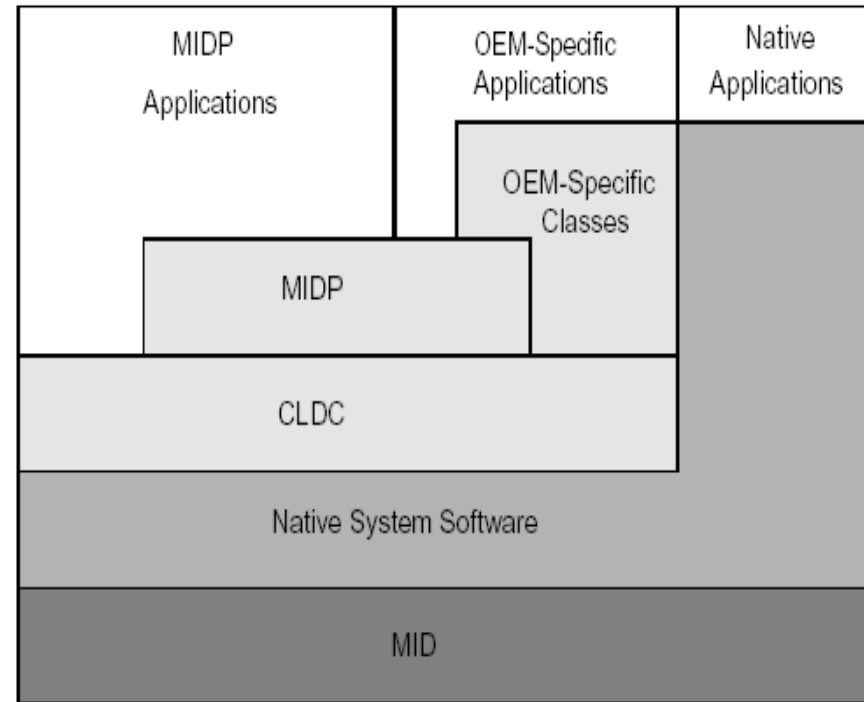
Configuraciones/Perfiles

□ Perfil:

- Definido a partir de una configuración.
- Define un grupo adicional de APIs.
- Librerías que permiten acceder a funcionalidades específicas del dispositivo:
 - GUI.
 - Capacidad de comunicación ampliada.
 - Almacenamiento persistente.

CLDC/MIDP

- CLDC: funcionalidades independientes del dispositivo (librerías generales y de comunicación).
- MIDP: funcionalidades específicas (GU comunicación ampliada, almacenamiento persistente)



Librerías

- J2ME contiene un subconjunto de la funcionalidad de la versión Java de sobremesa:
 - No todos los paquetes.
 - No todas las clases de un paquete.
 - No todas las funciones de una clase.

CLDC

□ Dos grupos de clases:

- Heredadas de J2SE. Encuadradas dentro de las librerías:
 - java.util
 - java.lang
 - java.io
- Específicas de esta configuración
 - javax.microedition.io

CLDC: java.lang

- Boolean
- Byte
- Character
- Class
- Integer
- Long
- Math
- Object
- *Runnable*
- Runtime
- Short
- String
- StringBuffer
- System
- Thread
- Throwable

CLDC: java.util

- Calendar
- Date
- *Enumeration*
- Hashtable
- Random
- Stack
- TimeZone
- Vector

CLDC: java.io

- ByteArrayOutputStream
- ByteArrayInputStream
- *DataOutput*
- *DataInput*
- DataInputStream
- DataOutputStream
- InputStream
- InputStreamReader
- OutputStream
- OutputStreamWriter
- PrintStream
- Reader
- Writer

CLDC: javax.microedition.io

- ❑ Marco necesario para establecer diferentes tipos de comunicación.
- ❑ Generic Connection Framework.
- ❑ La mayor parte son interfaces que representan los diferentes tipos de conexiones: http, socket, datagramas, serie, etc.

MIDP

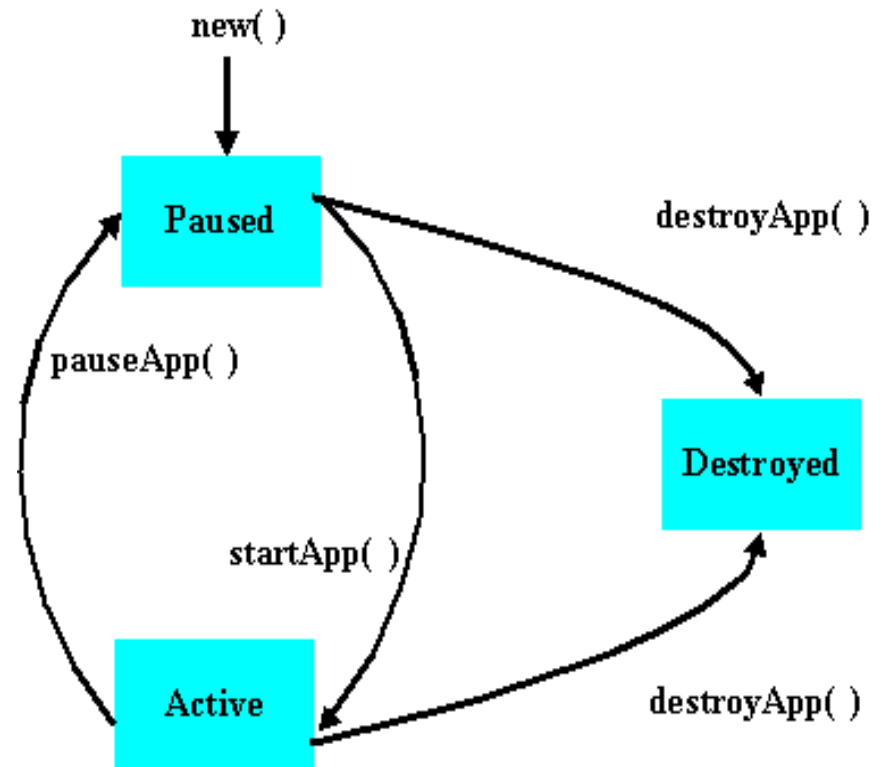
- javax.microedition.midlet
 - Define las aplicaciones MIDP y las interacciones entre la aplicación y el entorno en el que se ejecuta.
- javax.microedition.lcdui
 - Clases asociadas con GUI y el soporte de eventos.
- javax.microedition.rms
 - Mecanismos para el almacenamiento persistente.

MIDP (paquete midlet)

- Las aplicaciones MIDP se denominan MIDlets:
 - Deben heredar de la clase MIDlet (javax.microedition.MIDlet).
 - Deben implementar tres métodos:
 - startApp()
 - pauseApp()
 - destroyApp(boolean unconditional)

MIDP (ciclo de vida de un MIDlet)

- Start: Equivalente a main(). Se utiliza para comenzar la ejecución de la aplicación.
- Paused: la aplicación libera recursos y espera.
- Destroyed: la aplicación libera recursos y actividades, finalizando la aplicación.



GUI (Interfaz gráfica)

- ❑ No es un subconjunto de AWT o Swing.
- ❑ javax.microedition.lcdui
- ❑ Se compone de dos grupos de APIs:
 - De alto nivel:
 - Trabajan desde un punto de vista más conceptual \Rightarrow mayor grado de portabilidad.
 - No hay acceso a las características nativas del dispositivo.
 - De bajo nivel:
 - Control total del diseño de la pantalla (juegos).
 - Menor portabilidad.

Networking

- Las diferentes posibilidades de comunicación están representadas por 7 interfaces:
 - Connection
 - ContentConnection
 - DatagramConnection
 - InputConnection
 - OutputConnection
 - StreamConnection
 - StreamConnectionNotifier
- Todas las conexiones se crean de la misma manera:
 - Connection Connector.open(String s)

Networking

- Formato del String pasado como parámetro:
 - {protocol}:{target}[/params]
- Valores para protocol:
 - file
 - comm
 - socket
 - datagram
 - http
- Target ⇒ hostname, puerto, nombre de un archivo.
- Params ⇒ opcional (información adicional)

Networking

□ Ejemplos:

- `Connection c=Connector.open(http://www.javasoft.com)`
- `Connection c=Connector.open("socket://localhost:8080")`
- `Connection c=Connector.open("datagram://www.deusto.es:9000")`
- `Connection c=Connector.open("comm:0;baudrate=9000")`
- `Connection c=Connector.open("file:/foo.dat")`

□ Otros métodos de Connector:

- `open(String s,int mode)`
- `open(String s,int mode,boolean timeouts)`
- `DataInputStream openDataInputStream(String s)`
- `DataOutputStream openDataOutputStream(String s)`
- `InputStream openInputStream(String s)`
- `OutputStream openOutputStream(String s)`

Networking (sockets)

- ❑ No todos los móviles soportan este tipo de conexión.
- ❑ Envío de datos \Rightarrow `OutputConnection` \Rightarrow `OutputStream`
- ❑ Recepcion de datos \Rightarrow `InputConnection` \Rightarrow `InputStream`
- ❑ `StreamConnection` \Rightarrow recepción y envío de datos
- ❑ Hay que cerrar el socket al finalizar la aplicación.

Almacenamiento persistente

- Dos tipos de memoria:
 - Ejecución de las aplicaciones.
 - Almacenamiento persistente.
- RMS (Record Management System)
- RecordStore \Rightarrow tabla
- record \Rightarrow registro
- javax.microedition.rms

Paquetes opcionales

□ Web Services API:

- Abre nuevas posibilidades a J2ME:
 - Permite acceso remoto a web services mediante SOAP / XML.
 - Permite parsing XML.
- Muy importante dado el auge actual de XML.